UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/702,462	10/31/2000	Alan L. Davis	TI-30558	1089

23494 7590 10/20/2003

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	4

DATE MAILED: 10/20/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/702,462	DAVIS ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication app ars on the cover sheet with the correspondenc address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 October 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 October 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-21 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: #3. Declaration as received on 2/5/2001.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

4. The abstract of the disclosure is objected to because of the following minor informalities: Line 4 of the abstract includes a sentence fragment. Please reword it to be a proper sentence. In line 5, replace "in" with --an--. In lines 4 and 6, remove the reference labels assigned to the microprocessor and the functional units. In line 17, replace "instruction" with --instructions--. In line 18, replace "address" with --addresses--. Finally, remove "Figure 9" from the end of the abstract. Correction is required. See MPEP § 608.01(b).

5. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

6. The disclosure is objected to because of the following informalities: When referring to other applications within the specification (provisional, related, etc.), please include the serial number or patent number and remove the attorney docket number. On page 6, line 8, a

description of Fig.2 is given. However, this should be changed to explicitly reflect both Fig.2A and Fig.2B. On page 10, line 8, "Figure 2" should be changed to more accurately reflect Figures 2A and 2B. On page 20, line 11, replace "a address" with --an address--. On page 20, line 21, replace "502" with --503--. On page 21, line 8, replace "a address" with --an address--. On page 23, in Table 7, replace "u nit" with --unit--. Also, the format of the ADDKPC instruction does not match that of Fig.6A and Fig.6B. More specifically, the format in the table shows "src1, src2, dst" as the parameters while the format in Fig.6A and Fig.6B is "src1, dst, src2". On page 23, lines 6 and 19, the applicant should specify the meaning of "~31". On page 25, lines 7-9, replace each occurrence of "702" with --701--. On page 25, line 10, replace "7-4" with --702--.

Appropriate correction is required.

Drawings

7. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference sign(s) not mentioned in the description: In Fig.1, the examiner has been unable to find reference numbers 40b, 43, 82, 90, 100, and 102 within the specification. In Fig.2A and Fig.2B, the examiner has been unable to find reference numbers 12a,b, 14a,b, 16a,b, 18a,b, 40b, 102, 210, 211, 212, 213, 214, 220, 221, and 250 within the specification. In Fig.8, the examiner has been unable to find reference numbers 822(0), 824(1-3), and 825(0-1) within the specification. In Fig.11, the examiner has been unable to find reference number 15 within the specification. A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office

action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

8. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference characters "10" and "40" have both been used to designate the processor component within the cell phone. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

9. Claim 13 is objected to because of the following informalities: Please insert --of-- after "phase" in line 19 on page 36. Appropriate correction is required.

10. Claim 14 is objected to because of the following informalities: Please remove the reference numbers for each of the components listed in claim 14. Appropriate correction is required.

11. Claim 17 is objected to because of the following informalities: Please change "Claim15" to --Claim 15-- (note the space). Appropriate correction is required.

12. Claim 18 is objected to because of the following informalities: Please insert --a-- before "digital" in line 26 on page 37. Appropriate correction is required.

13. Claim 20 is objected to because of the following informalities: Please insert --of-- after "step" in line 9 on page 38. In addition, as currently written, claim 20 is dependent on itself. Therefore, the dependency of this claim should be modified. For purposes of this examination, the examiner will assume the applicant meant for claim 20 to be dependent on claim 19, just as

claim 11 is dependent on claim 10 (which are similar claims). Appropriate correction is required.

Claim Rejections - 35 USC § 102

14. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

15. Claims 1, 4, 7, 9-10, 15, and 18-19 are rejected under 35 U.S.C. 102(a) as being anticipated by Applicant's Admitted Prior Art (herein referred to as AAPA).

16. Referring to claim 1, AAPA has taught a digital processing system comprising a microprocessor (see page 20, lines 8-9), wherein the microprocessor is operable to perform a method for calling a subroutine, the method comprising the steps of:

a) branching to the subroutine by executing a first instruction to provide an address of the subroutine. See Fig.5, instruction 501, and page 20, lines 9-16.

b) calculating a return address by executing a second instruction to determine a return address. See Fig.5, instruction 503, and page 20, lines 17-26. When this second instruction completes, a return address will have been formed that is relative to zero.

17. Referring to claim 4, AAPA has taught a system as described in claim 1. AAPA has further taught that the second instruction is executed during a delay slot associated with the first instruction. See Fig.5 and page 20, line 27, to page 21, line 3.

18. Referring to claim 7, AAPA has taught a system as described in claim 1. AAPA has further taught during the step of calculating a return address the return address is remotely associated with the second instruction. As described in the rejection of claim 1, the second instruction will form a return address. Therefore, this address is remotely associated with the second instruction because the return address is determined by the second instruction.

19. Referring to claim 9, AAPA has taught a digital processing system comprising a microprocessor (see page 20, lines 8-9), wherein the microprocessor is operable to perform a method for forming a relative address, the method comprising the steps of:

a) fetching a sequence of instructions in response to address locations provided by a program counter. It is inherent that instructions are fetched based on an address within the program counter. Without a program counter, instructions would not be fetched and programs would not be executed.

b) executing a first instruction of the sequence of instructions by using a first address value provided by the program counter as a source operand. See Fig.5, instruction 501, and page 20, lines 9-16. Note that a displacement value supplied by the branch instruction is added to the program counter's contents.

20. Referring to claim 10, AAPA has taught a digital system as described in claim 9. AAPA has further taught that the step of executing a first instruction comprises the step of combining a displacement value provided by the first instruction with the first address value provided by the program counter. As discussed in the rejection of claim 10 above, see Fig.5, instruction 501, and page 20, lines 9-16. Note that a displacement value supplied by the branch instruction is added to the program counter's contents.

21. Referring to claim 15, AAPA has taught a method for calling a subroutine in a digital processing system comprising a microprocessor, the method comprising the steps of:

a) branching to the subroutine by executing a first instruction to provide an address of the subroutine. See Fig.5, instruction 501, and page 20, lines 9-16.

b) calculating a return address by executing a second instruction to determine a relative return address. See Fig.5, instruction 503, and page 20, lines 17-26. When this second instruction completes, a return address will have been formed that is relative to zero.

22. Referring to claim 18, AAPA has taught a method for forming a relative address in digital processing system comprising a microprocessor, the method comprising the steps of:

a) fetching a sequence of instructions in response to address locations provided by a program counter. It is inherent that instructions are fetched based on an address within the program counter. Without a program counter, instructions would not be fetched and programs would not be executed.

b) executing a first instruction of the sequence of instructions by using a first address value provided by the program counter as a source operand. See Fig.5, instruction 501, and page 20, lines 9-16. Note that a displacement value supplied by the branch instruction is added to the program counter's contents.

23. Referring to claim 19, AAPA has taught a method as described in claim 18. AAPA has further taught that the step of executing a first instruction comprises the step of adding a displacement value provided by the first instruction to the first address value provided by the program counter. As discussed in the rejection of claim 10 above, see Fig.5, instruction 501, and

page 20, lines 9-16. Note that a displacement value supplied by the branch instruction is added to the program counter's contents.

24. Claims 1-3, 6-7, and 15-17 are rejected under 35 U.S.C. 102(b) as being anticipated by Iwao, U.S. Patent No. 4,799,151.

25. Referring to claim 1, Iwao has taught a digital processing system comprising a microprocessor, wherein the microprocessor is operable to perform a method for calling a subroutine, the method comprising the steps of:

a) branching to the subroutine by executing a first instruction to provide an address of the subroutine. See Fig.5, and column 1, lines 15-21, and note that when a BAL (branch and link) instruction is executed, an address to a subroutine is provided.

b) calculating a return address by executing a second instruction to determine a relative return address. See Fig.5, and note that at the end of a subroutine, a return instruction is executed, whereby the return PC is added to a constant provided by the return instruction, so that a target address is formed. For instance, from Fig.5, the initial return address after the BAL instruction is address 521. If the return at the end of block S3 executes, the value 5 is added to the return PC to obtain the address 526 (therefore, the return would cause a return back to block A3 of the program).

26. Referring to claim 2, Iwao has taught a system as described in claim 1. Iwao has further taught that the step of calculating a return address comprises the step of adding a relative displacement value provided by the second instruction to a program address value associated with the second instruction. Again, see Fig.5 and note that the return program address is set after

the BAL instruction and is associated with a return instruction (in this case, the return program address is 521). The second instruction (return) will optionally add a displacement value to the return program address. This can be seen at the end of block S3 when the return instruction adds 5 to the return program address 521, resulting in a return address of 526.

27. Referring to claim 3, Iwao has taught a system as described in claim 2. Iwao has further taught that the step of calculating a return address further comprises the step of storing the return address in a general-purpose register of the microprocessor. See Fig.3, component 19, and column 4, lines 16-18.

28. Referring to claim 6, Iwao has taught a system as described in claim 1. Iwao has further taught that during the step of calculating a return address, a plurality of second instructions are conditionally executed in response to a predicate value such that the return address is responsive to the predicate value. See Fig.5 and note that a first return or second return instruction will be conditionally executed based on the predicate value of the determination. If the value is true (yes), the return address will be calculated according to the return instruction at the end of block S3. On the other hand, if the value is false (no), the return address will be calculated according to the return instruction at the end of block S2.

29. Referring to claim 7, Iwao has taught a system as described in claim 1. Iwao has further taught during the step of calculating a return address the return address is remotely associated with the second instruction. As described in the rejection of claim 1, the second instruction will form a return address. Therefore, this address is remotely associated with the second instruction because the return address is determined by the second instruction.

30. Referring to claim 15, Iwao has taught a method for calling a subroutine in a digital processing system comprising a microprocessor, the method comprising the steps of:

a) branching to the subroutine by executing a first instruction to provide an address of the subroutine. See Fig.5, and column 1, lines 15-21, and note that when a BAL (branch and link) instruction is executed, an address to a subroutine is provided.

b) calculating a return address by executing a second instruction to determine a relative return address. See Fig.5, and note that at the end of a subroutine, a return instruction is executed, whereby the return PC is added to a constant provided by the return instruction, so that a target address is formed. For instance, from Fig.5, the initial return address after the BAL instruction is address 521. If the return at the end of block S3 executes, the value 5 is added to the return PC to obtain the address 526 (therefore, the return would cause a return back to block A3 of the program).

31. Referring to claim 16, Iwao has taught a method as described in claim 15. Iwao has further taught that the step of calculating a return address comprises the step of adding a relative displacement value provided by the second instruction to a program address value associated with the second instruction. Again, see Fig.5 and note that the return program address is set after the BAL instruction and is associated with a return instruction (in this case, the return program address is 521). The second instruction (return) will optionally add a displacement value to the return program address. This can be seen at the end of block S3 when the return instruction adds 5 to the return program address 521, resulting in a return address of 526.

32. Referring to claim 17, Iwao has taught a method as described in claim 15. Iwao has further taught the step of calculating a return address further comprises the step of storing the

return address in a general-purpose register of the microprocessor. See Fig.3, component 19, and column 4, lines 16-18.

Claim Rejections - 35 USC § 103

33. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

34. Claims 5, 12, and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, as applied above, in view of Hennessy and Patterson, Computer Architecture - A Quantitative Approach, 2nd Edition, 1996 (herein referred to as Hennessy).

35. Referring to claim 5, AAPA has taught a system as described in claim 1. AAPA has not explicitly taught that the second instruction is executed before the first instruction. However, Hennessy has shown on page 169 that an instruction that is independent of the branch instruction can be executed either before the branch instruction or after the branch instruction (in the delay slot). See Figure 3.28(a). The purpose of executing a second instruction after the first instruction (in a delay slot) is to perform optimization. However, this optimization is not required. Therefore, since a "branch-independent" can be executed either before or after the branch instruction, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify AAPA to execute the second instruction before the first instruction.

36. Referring to claim 12, AAPA has taught a system as described in claim 10.

a) AAPA has not explicitly taught an instruction execution pipeline having a plurality of stages, the program counter being associated with a fetch stage of the instruction execution pipeline, and a first functional circuit associated with an execution phase of the instruction execution pipeline. However, Hennessy has shown a basic pipeline structure which includes all of the aforementioned components. See Figure 3.1 on page 130 and Figure 3.4 on page 134. Note that the pipeline is 5 stages where one of the stages is a fetch stage which is associated with a program counter (PC). In addition, the execution stage is associated with a first functional unit (ALU), as is well known in the art. Implementing such a pipeline allows for an increase in instruction level parallelism, which in turn yields higher throughput and faster execution speeds. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a pipeline as taught by Hennessy into the system of AAPA in order to increase throughput. Also it should be noted that for an instruction to be fetched in a fetch stage, a program counter must be associated with the fetch stage in order to provide a fetch address. In addition, it should also be realized that for in order to perform some operation (which occurs in the execution stage), a functional unit must exist.

b) Furthermore, AAPA has not taught FIFO circuitry connected to receive address values from the program counter, the FIFO circuitry operable to delay each address value received from the program counter, wherein the first functional circuit is operable to add the displacement value provided by the first instruction to a first delayed address value provided by the program counter. However Hennessy has taught FIFO circuitry which delays a PC value until it is finally added to a displacement value by a first functional unit. See Figure 3.4 on page 134. It should be noted that the group of pipeline registers (IF/ID, ID/EX, EX/MEM, and MEM/WB) form a FIFO,

where data is passed from one register to the next starting with IF/ID. As discussed in the last sentence of the caption of the Figure, the PC information is carried from left to right (in a FIFO fashion) by these registers for branch instructions. The delayed PC of the corresponding branch will arrive at the ALU in the EX stage where it will be added to a displacement value provided by the instruction fetched from the instruction memory. Figure 3.22 of Hennessy on page 163 shows that this same concept can be performed with less delay as well (where the displacement is added one cycle sooner). As described in part (a) above, Hennessy has disclosed that this is circuitry found in a pipelined system and implementing such a pipeline will increase throughput and execution speeds due to the overlapping execution of instructions. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to include this FIFO circuitry which delays a PC value.

37. Referring to claim 21, AAPA has taught a microprocessor that is operable to execute a first type of instruction that performs a specified operation and also directs that a selectable number of virtual no-operation (NOP) instructions be executed after executing the first type instruction (see Fig.5 and page 20, line 8, to page 21, line 3), wherein:

a) it has been determined that a first instruction of the first type of instruction is to be executed in a delay slot of a first branch type instruction. See Fig.5, instruction 503.

b) it has been determined that a second instruction of a second type of instruction is to be executed in a delay slot of the first branch type instruction. See Fig.5, instruction 502.

c) an instruction sequence has been arranged such that the first instruction is executed after the second instruction. See Fig.5 and note that the first instruction (503) is executed after the second instruction (502).

d) a remaining number of delay slots of the first branch type instruction has been determined. See Fig.5 and note that since a branch type instruction has 5 delay slots (according to page 20), and two of the slots have been filled with instructions 502 and 503, the amount of remaining slots open is three. These slots are then filled with a number of NOP instructions (as shown in Fig.5).

e) AAPA has not taught inserting a value in the first instruction to direct that a selectable number of virtual NOP instructions be executed after executing the first type instruction, wherein the selectable number of NOPs is the remaining number of unused delay slots of the first branch instruction. However, TI has taught the concept of combining multiple instructions into a single instruction. See page 3-7 and note that the LTA instruction combines the LT and APAC instructions. In general, the integration of multiple components is not given patentable weight or would have been an obvious improvement. One of ordinary skill in the art would have recognized that the explicit NOP instruction (504) of AAPA could be combined with the first instruction (503) in order to reduce the amount of explicit instructions that are executed. This would in turn increase execution speeds. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to insert a NOP value into the first instruction (combine instructions) of AAPA in order to specify the amount of virtual NOPs that are to be executed after the branch.

f) AAPA has not explicitly taught that the above determining, arranging, and inserting steps form a method of operating a compiler. However, Hennessy has taught that it is the job of a compiler to optimize the code before it is executed. See page 169. Hennessy has shown that rearranging code into delay slots of branches is the job of the compiler. This allows the code to be fully

optimized before execution begins, thereby allowing the microprocessor to run in an efficient manner. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the above steps be a method for operating a compiler.

38. Claims 8, 11, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, as applied above, in view of Texas Instruments, TMS32010 User's Guide, 1983 (herein referred to as TI).

39. Referring to claim 8, AAPA has taught a system as described in claim 1. AAPA has further taught the step of filling a number of delay slots associated with the first instruction in an instruction execution pipeline of the microprocessor by executing a number of virtual no-operation (NOP) instructions. See Fig.5, instruction 504. AAPA has not taught that the number of virtual NOPs is specified by the second instruction. However, TI has taught the concept of combining multiple instructions into a single instruction. See page 3-7 and note that the LTA instruction combines the LT and APAC instructions. In general, the integration of multiple components is not given patentable weight or would have been an obvious improvement. One of ordinary skill in the art would have recognized that the explicit NOP instruction (504) of AAPA could be combined with the second instruction (503) in order to reduce the amount of explicit instructions that are executed. This would in turn increase execution speeds. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the second instruction of AAPA specify the amount of virtual NOPs that are to be executed in the branch's delay slots.

40. Referring to claim 11, AAPA has taught a system as described in claim 10. AAPA has further taught providing a number of virtual NOP instructions for execution after the step of executing the first instruction. See Fig.5, instruction 504. AAPA has not taught that the number of virtual NOPs is provided during the execution of the first instruction. However, TI has taught the concept of combining multiple instructions into a single instruction. See page 3-7 and note that the LTA instruction combines the LT and APAC instructions. In general, the integration of multiple components is not given patentable weight or would have been an obvious improvement. One of ordinary skill in the art would have recognized that the explicit NOP instruction (504) of AAPA could be combined with the first instruction in order to reduce the amount of explicit instructions that are executed. This would in turn increase execution speeds. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the first instruction of AAPA specify the amount of virtual NOPs that are to be executed after the branch.

41. Referring to claim 20, AAPA has taught a system as described in claim 19. AAPA has further taught providing a number of virtual NOP instructions for execution after the step of executing the first instruction. See Fig.5, instruction 504. AAPA has not taught that the number of virtual NOPs is provided during the execution of the first instruction. However, TI has taught the concept of combining multiple instructions into a single instruction. See page 3-7 and note that the LTA instruction combines the LT and APAC instructions. In general, the integration of multiple components is not given patentable weight or would have been an obvious improvement. One of ordinary skill in the art would have recognized that the explicit NOP instruction (504) of AAPA could be combined with the first instruction in order to reduce the

amount of explicit instructions that are executed. This would in turn increase execution speeds. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the first instruction of AAPA specify the amount of virtual NOPs that are to be executed after the branch.

42. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of Hennessy, as applied above, and further in view of Sharangpani et al., U.S. Patent No. 6,237,077 (herein referred to as Sharangpani).

43. Referring to claim 13, AAPA in view of Hennessy has taught a system as described in claim 12.

a) AAPA in view of Hennessy has not taught a plurality of functional units associated with the execution phase the instruction execution pipeline, wherein the instruction execution pipeline receives a fetch packet containing a plurality of instructions associated with each address value provided by the program counter, and wherein a dispatch stage of the pipeline is operable to provide an execution packet that spans two or more fetch packets. However, Sharangpani has taught such a system. See Fig.2 and notice the fetch packet which contains a plurality of instructions. And, from Fig.3, it has been taught that the plurality of instructions within the fetch packets are accommodated by a plurality of functional units (M0, M1, I0...BR2). Finally, Sharangpani has taught a dispatch stage (Fig.3) where execution packets are formed. It should be realized from Fig.2 that each fetch packet includes a stop field. The stop field of a first packet indicates whether or not inter-group dependencies exist between the first packet and the next packet. See column 4, lines 4-5. More specifically, if the stop field indicates that no group

boundary exists between fetch packets, then instructions from multiple fetch packets are considered to be in the same group (thereby creating a larger execution packet), and instructions in the same group are executed in parallel. With this in mind, it can be seen that the size of an execution packet can vary (it can grow if multiple fetch packets are in the same group (denoted by the stop bit). Dispatch circuitry (Fig.3, components 330 and 340) will check the stop field and for each fetch packet, and route the instructions appropriately. See column 7, lines 59-62. This system and setup, in general, allows multiple instructions to be executed in parallel during the same cycle by multiple functional units, as opposed to executing a single instruction per cycle with a single functional unit. As a result, since more instructions would be executed in parallel, throughput would be higher and it would take less time to execute a program. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify AAPA in view of Hennessy in view of Sharangpani to speed up execution.
b) Recall from the rejection of claim 12 that Hennessy has taught that the FIFO circuitry will provide a delayed address value associated with a branch instruction. Therefore, it follows that if the branch instruction is in a fetch packet, then that delayed address value is associated with a fetch packet (this corresponds to the final paragraph of claim 13).

44. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA in view of in view of Haataja, U.S. Patent No. 6,137,836.

45. Referring to claim 14, AAPA has taught a digital system as described in claim 1. AAPA has not taught that the digital system is a cellular telephone comprising the components set forth in claim 14. However, Haataja has taught a cellular telephone comprising:

a) an integrated keyboard connected to the CPU via a keyboard adapter. See Fig.8, component 72.

b) a display, connected to the CPU via a display adapter. See Fig.8, component 36.

c) radio frequency (RF) circuitry connected to the CPU. See Fig.8, component 56, and column 7, lines 6-11.

d) an aerial connected to the RF circuitry. See Fig.8, component 54.

It should be realized that AAPA has taught a system that includes operations that increase the functionality of the system. A person of ordinary skill in the art would have recognized that an improved processor (with more functionality) would lead to the overall improvement of the device in which it is embedded. As shown in Fig.8 of Haataja, and, as is well known in the art, cellular telephones are controlled by some sort of processor. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the incorporate the digital system of AAPA into a cell phone, as taught by Haataja, in order to improve the overall performance of the cell phone.

Conclusion

46. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Brown et al., U.S. Patent No. 5,958,044, has taught a system with multi-cycle NOPs.

Rogers et al., Supporting Dynamic Data Structures on Distributed-Memory Machines, March 1995, has taught the execution of a single instruction during a delay slot of a branch which modifies a return address and stores the return address in a register (pg.259).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

DJH
David J. Huisman
October 9, 2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100